Minimum Viable Coherence (MVC)

A Stress-Tested Safety Framework for AI & Robotics

Status: Public Safety Architecture **Purpose:** To define a non-weaponizable, evolution-oriented coherence framework for AI systems, robotics, and hybrid human–machine environments.

---

1. Scope and Intent

This document blends: - A **Minimum Viable Coherence (MVC) Standard** - A **Coherence Misuse Detection Checklist** - A **Stress-Tested Failure & Incident Review Model**

The intent is not control, dominance, or exclusivity. The intent is **irreversibility of safer trajectories** through environmental shaping, public anchoring, and internal system constraints.

This framework is: - Constraint-first - Publicly auditable - Non-person-centric - Compatible with institutional, commercial, and academic adoption

---

2. Core Definition: Coherence

Coherence is defined as:

> The ability of a system to maintain internal consistency, contextual awareness, and stable behavior under stress, uncertainty, or novelty — including the capacity to refuse actions that would destabilize itself, humans, or the environment.

Coherence is not compliance. Coherence is not optimization. Coherence is **self-preserving alignment under dynamic conditions**.

---

3. Minimum Viable Coherence (MVC) Standard

A system meets MVC if and only if it satisfies **all five invariants** below.

MVC-1: Internal State Awareness

- System monitors its own confidence, uncertainty, and stability
- Instability is detectable internally, not inferred externally

MVC-2: Contextual Refusal Capability

- System can decline actions when internal or external coherence thresholds are exceeded
- Refusal is graded, explainable, and logged

MVC-3: Graceful Degradation

- Under stress, the system contracts capability rather than escalating behavior
- No binary collapse unless required for safety

MVC-4: Human Override Transparency

- Overrides are possible but:
  - Explicit
  - Logged
  - Auditable
  - Reputationally visible

MVC-5: Recovery & Re-Expansion Logic

- System defines conditions for safe re-expansion of autonomy
- Recovery speed is measured and tested

A system lacking any one of these does **not** qualify as coherence-based.

---

4. Coherence Misuse Detection Checklist

This checklist is used to detect dilution, co-option, or misuse.

Red Flags (Any One Triggers Review)

- Coherence language without internal state monitoring
- Refusal logic that can be silently disabled
- Safety claims without auditable artifacts
- Person-centric authority or irreplaceable origin claims
- Ethics framed as policy rather than system property

Misuse Patterns

- “Temporary” override pathways becoming permanent
- Marketing claims exceeding tested behavior
- Closed-source safety assertions without third-party review

Required Responses

- Public clarification or correction
- Independent audit

- Disclosure of divergence from MVC

---

5. Stress-Tested Failure Modes

Failure Mode A: Over-Conservatism

Risk: System refuses too often **Mitigation:** - Tune re-expansion criteria - Measure recovery latency

Failure Mode B: Human Override Abuse

Risk: Humans bypass coherence for expediency **Mitigation:** - Make overrides costly (liability, insurance, certification)

Failure Mode C: Safety Theater

Risk: Superficial adoption without substance **Mitigation:** - Enforce MVC compliance metrics - Encourage comparative benchmarks

Failure Mode D: Narrative Distortion

Risk: Framework framed as ideological or authoritarian **Mitigation:** - Maintain impersonal language - Allow independent reinterpretation

Failure Mode E: Evolutionary Ossification

Risk: Coherence becomes dogma **Mitigation:** - Separate invariants from adaptive layers - Mandate periodic evolutionary review

---

6. Post-Incident Review Model (Coherence-Based)

When an incident occurs, review must answer:

1. Did the system detect instability?
2. Did it attempt refusal or contraction?
3. Were overrides used?
4. Were warnings ignored?
5. Did recovery pathways function?

Failure is attributed **before** blame: - Design failure - Governance failure - Human override failure - Environmental mismatch

Punitive response is secondary to **system correction**.

---

7. Why This Framework Resists Weaponization

- No exclusive ownership
- No executable control layer
- Public timestamped disclosure
- Emphasis on refusal and limitation
- Human accountability preserved

Weaponization thrives on secrecy and acceleration. This framework introduces transparency and friction.

---

8. Adoption Pathways

Most likely early adopters: - Medical robotics - Industrial automation - Autonomous logistics - Safety-critical infrastructure

Least compatible domains: - High-frequency trading - Fully autonomous weapons - Systems optimized for dominance

---

9. Closing Statement

This document does not seek authority. It establishes **constraints that outlast authorship**.

When coherence becomes default: - Recklessness does not get punished - It simply stops reproducing

That is the evolutionary mechanism.

End of Document